Navigation of a Mobile Robot Using a Virtual Potential Field and Artificial Neural Network

Mohsen Ashourian
Department of Engineering, Majlesi Branch, Islamic Azad University Majlesi, Iran
Email: ashourian@iaumajlesi.ac.ir

Abstract

Mobile robot navigation is one of the basic problems in robotics. In this paper, a new approach is proposed for autonomous mobile robot navigation in an unknown environment. The proposed approach is based on learning virtual parallel paths that propel the mobile robot toward the track using a multi-layer, feed-forward neural network. For training, a human operator navigates the mobile robot in some different paths in the environment. Both of human operator navigating data and virtual parallel paths train the neural network. The neural network is able to map the coordinate of a position to a mobile robot orientation and velocity. After training, the mobile robot can plan a track between start and target position without the need of any human operator. When the environment surrounding the mobile robot is unknown, sensors are used to detect obstacles and avoid collision. The simulated mobile robot is equipped with a rangefinder sensor. The simulation shows promising results and high speed for real-time implementation in unknown and partially dynamic environments.

Keywords: Mobile robot, Navigation, Neural networks

1- Introduction

In the last few years, many researchers have been interested in mobile robots. Nowadays, significant improvements have been achieved in the realm of kinematics and dynamics of mobile robots. However, one of the main issues is mobile robot navigation in an unknown environment. A suitable solution depends on accurate understanding of the environment using sensors and taking advantage of advanced techniques to navigate the mobile robot toward the target position and avoid obstacles [1]. Navigation is a critical problem encountered in the operation of a mobile robot. It is necessary to have capability of detecting environmental conditions and analyzing sensory data to coordinate the mobile robot components in order to have a collision free trajectory [2]. Obstacles could be in diverse shapes and sizes; also, some of them are allowed to move. Important features that distinguish navigation algorithms are whether the environment is known or unknown, and whether it is static or dynamic [3]. Known environments are those in which the mobile robot knows how to coordinate the obstacles, and the target point is clearly defined. In unknown environments, the mobile robot does not have any previous knowledge about the environment, which means that it does not know if there are obstacles, where the obstacles are, and what is the accurate location of the target position. Therefore, the mobile robot must use sensors to detect the environmental information and can only sense information within the range of these sensors. There are two conditions that make environments dynamic. The first condition is that the target moves continuously during the navigation. Many mobile robot navigation algorithms have been proposed for such environments [4, 5].

A characteristic of these algorithms is that the target point must always be in the mobile robot's field of view. The second condition is that there are moving obstacles appearing randomly in the known or unknown portions of the environment. Navigation algorithms in such an environment are often based on the combination of a navigation algorithm for static environments and a collision-avoidance strategy. Navigation algorithms are divided into two categories. The first, attempts to find the best possible path. Examples of successful methods for this kind of path planning includes the traditional artificial potential field, the A* and D* algorithms [6], and many others. The second tries to find an acceptable path in a short time; Genetic Algorithm, Neural Networks, and Fuzzy Logic are classified in the second category [7]. Many researchers have been investigating the navigation of mobile robots using artificial neural network (ANN) in an unknown and dynamic environment. As a research, a recurrent neural network (RNN) model for mobile robot navigation was proposed. This model has been usedas a kind of supervised learning method used to train the mobile robot. The learning process in RNNs will spend a lot of time [8]. In other case, a multi-layer, feed-forward ANN is used to navigate a mobile robot in a 2D environment in aid of a human operator. The model, received data from a camera, processed them using ANN and finally sent command to mobile robot actuators [9]. A research has done on Hopfield neural networks (HNN) in mobile robot navigation. HNNs are not able to learn. Hence, they are not suitable for dynamic environments [10]. In [11], a model was introduced using multi-layer, feed-forward ANN to navigate a car-like mobile robot. The presented model was dependent on the mobile robot kinematics [11]. In [12], a hybrid approach based on a multi-layer, feed-forward ANN and fuzzy logic is proposed. The implemented ANN has two inputs, which are orientation and distance between the mobile robot and the obstacle, and two outputs, which are velocity and orientation. In this paper, a new approach is proposed. The approach based on learning virtual parallel paths that propel the mobile robot toward the track. This approach is suitable for the real-time navigation of a mobile robot in unknown and partially dynamic environments.

The ANN is trained by Levenberg-Marquardt, back-propagation algorithm under a supervised learning method. Results are obtained from software simulation. The mobile robot learns when to change orientation and velocity in current position to get target position. The simulation results show that even in environments with complex obstacles the mobile robot can move along a globally optimal or near-optimal path and arrive at the destination avoiding collision by using proposed new approach.

In Section 2, the simulating environment is described and the related definitions and structures are given. In Section 3, the new navigation approach is proposed. In Section 4, computer simulations and comparisons are given. Finally, the conclusions are presented.

2- Enviromenttal Definitions

Although kinematics of a mobile robot is important in the navigation, in general, the navigation can be studied independently . In this case, a simulated four-wheel, mobile robot is considered. The graphical design of the environment is shown in Figure 1. The environment is a flattened 2D plane with no slopes. It is defined as a plane size of 300×300 blocks. The coordinates of the plane's top-left and bottom-right corners, respectively are (-150, -150) and (+150, +150). This environment has the reciprocating crossings shown in white color. Obstacles are in gray color. There are 20 obstacles in the environment. Also there are 30 four-way or three-way intersections. Some crossings between the two intersections are direct and some of them are turned to the left or right. The simulated mobile robot is equipped with four actuators for each wheel, a laser rangefinder sensor, a digital camera, a GPS, and a programmable controller. Understanding the surrounding environment of the mobile robot can be done with the processing of sensory data. The digital camera is installed on top of the mobile robot and a laser rangefinder is used in front of the mobile robot. The mobile robot can find the current location of itself in the environment. The mission of the controller is analyzing the sensory data and sending the necessary commands to actuators in order to move safely and have a collision free trajectory while the mobile robot achieved the target position.

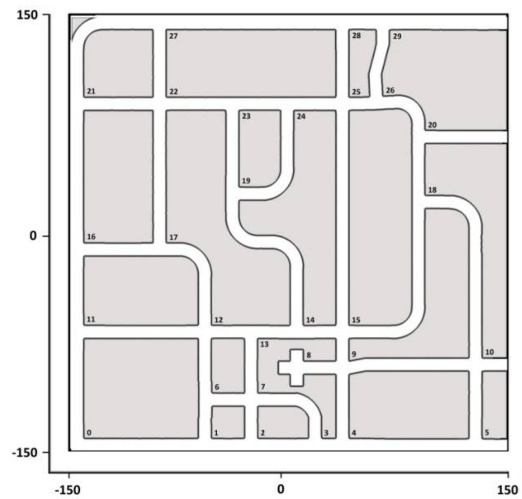


Fig.1. Schematic view of the simulated environment

3- The Proposed Approatch

3.1- First part

All The problem of navigation in an unknown and dynamic environment is solved commonly by heuristic methods [13]. These methods cannot guaranty to find the best solution, but they try to find a path in a reasonable time [14]. In this approach, a multi-layer, feed-forward ANN as a kind of heuristic method ,is used for navigating the mobile robot.

In proposed approach, inputs of the ANN are the coordinates of the current and target position of the mobile robot. As the mobile robot moves, the coordinates of the current position are sampling in a period of the time. The input vector has four elements. Inputs are defined in the range of [+150, -150]. Velocity and orientation of the mobile robot are output elements of the ANN.

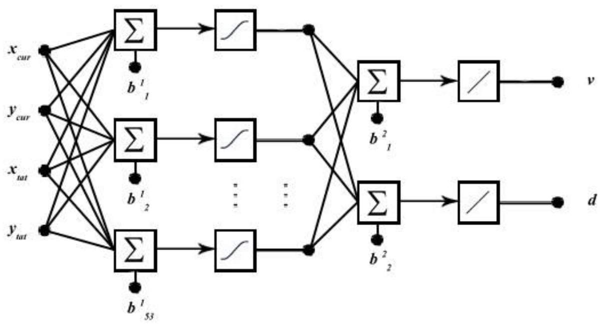


Fig.2. Schematic representation of the proposed model of the ANN

Figure 2 shows the architecture of the purposed model of the ANN. The velocity with positive value indicates that the mobile robot has to move forward and negative value indicates moving backward. The orientation is the angular deviation from the straight path. The orientation with negative value means turning the mobile robot left and positive value means turning right. The mobile robot orientation is defined in the range of [+0.4, -0.4] in radians. To improve the performance of the ANN, the velocity element is increased by 150 units and the orientation element (D_{real}) is adjusted based on Equation below:

$$D_{scaled} = (D_{real} \times 100) + 0.4 \tag{1}$$

D_{scaled} is the orientation of the mobile robot. When in motion, the mobile robot gets off track; it should attempt to return to the right path. Input matrix X has four rows. Two rows for the current position coordinates of the mobile robot (c_n^s , r_n^s), and two rows for the coordinates of the target position (c_n^d , r_n^d). Output matrix Y consists of two rows which are the velocity and the orientation of the mobile robot (Equations 2, 3).

$$X = \begin{bmatrix} r_1^s & r_2^s & r_3^s & ..... & r_n^s \\ c_1^s & c_2^s & r_3^s & ..... & c_n^s \\ r_1^d & r_2^d & r_3^d & ..... & r_n^d \\ c_1^d & c_2^d & r_3^d & ..... & c_n^d \end{bmatrix} \tag{2}$$

$$Y = \begin{bmatrix} D_{(r_1,c_1)} & D_{(r_2,c_2)} & ..... & D_{(r_n,c_n)} \\ V_{(r_1,c_1)} & V_{(r_2,c_2)} & ..... & V_{(r_n,c_n)} \end{bmatrix} \tag{3}$$

Pair of current and target position coordinates can be mapped to the velocity and the orientation by the ANN model. Determining an appropriate number of the neurons in a hidden layer is the way to achieve the optimal ANN output. The hidden layer of the ANN has a tangent sigmoid activation function and the output layer has a linear activation function.

In training process, a human operator handles the mobile robot to change the velocity and the orientation to achieve the target position. Obstacle avoidance is based on virtual potential field method. The software We bots 6.4.3is used to simulate the navigation of the mobile robot and results are obtained using MATLAB 2010. To evaluate the proposed approach, more than 50 tests were conducted. In this case, the mobile robot is trained for five different paths. Each of these paths is chosen in such a way that has some common intersections(Figure 3). Followed paths are shown in the set of intersection numbers that have passed:

P_1 = {1, 2, 3, 4, 9, 15, 18, 20}
P_2 = {18, 10, 15}
P_3 = {1, 2, 3, 4, 9, 8}
P_4 = {1, 2, 3, 4, 5, 10, 18, 20, 26, 29, 28, 27, 21, 16, 10, 0}
P_5 = {3, 7, 13, 12, 17, 22, 23, 19, 14}

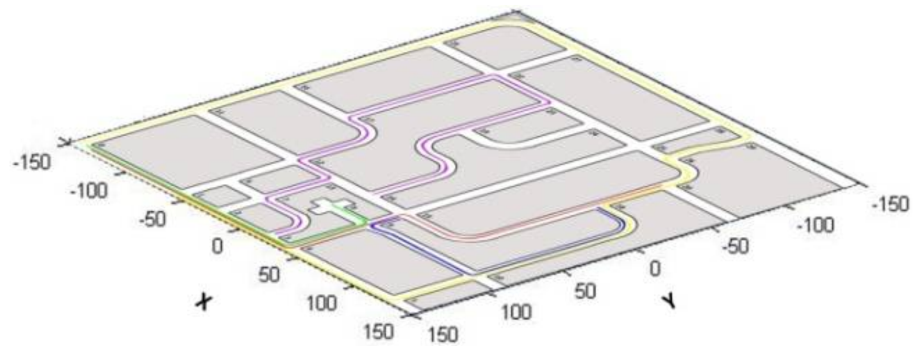


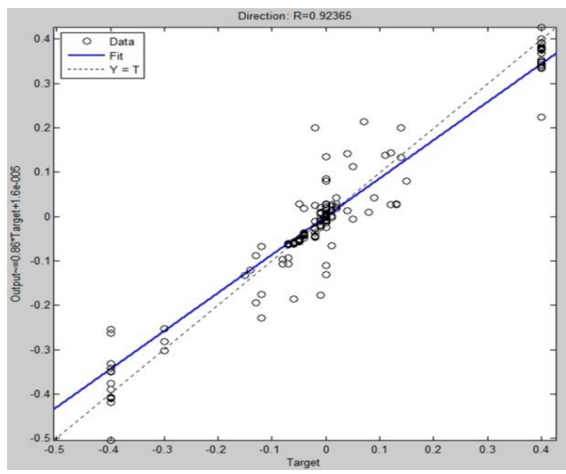
Fig.3. Training paths that the mobile robot has been learned by the human operator. paths are in different colors. P1, Red. P2, Blue. P3, Green. P4, Yellow. P5, Purplecaption …

3.1- Hidden layer neurons

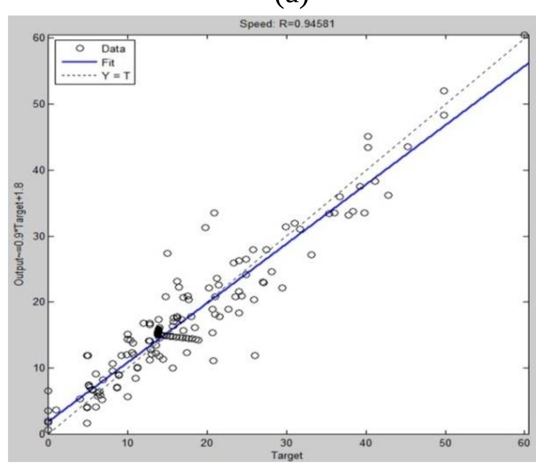
It is necessary to determine the number of hidden layer neurons for optimizing performance of the proposed model. In the first step, the total numbers of 170 epochs are considered. 153 epochs are used to train the ANN and 17 epochs for testing the ANN. Selected epochs are according to the different situations of the mobile robot. Second, the ANN is implemented with different number of neurons for hidden layer. Table 1 shows experimental related results. As it can be seen, output accuracy of the ANN is checked with start of 28 neurons in the hidden layer. Both the mobile robot velocity and orientation show a weak performance.

Table .1. The ANN performance for different number of hidden neurons

Num. of Neurons	Output accuracy		Output error	
	Orientation	Velocity	Orientation	Velocity
28	0.65804	0.80307	0.34196	0.19693
32	0.77029	0.86983	0.22971	0.13017
44	0.90665	0.91243	0.09335	0.08757
53	0.92365	0.94581	0.07635	0.05419
61	0.84632	0.89121	0.15368	0.10879



(a)



(b)

Fig.4. The actual output of the ANN versus the expected target values: (a) the velocity (b) the orientation

In some stages, output accuracy is improved by increasing the number of hidden layer neurons. The best performance is appeared in the ANN with 53 neurons of the hidden layer. 0.92365 for the orientation and 0.94581 for the velocity are suitable in this solution. Regression plot of the actual output of the ANN and the expected target values can be seen in Figure 4.

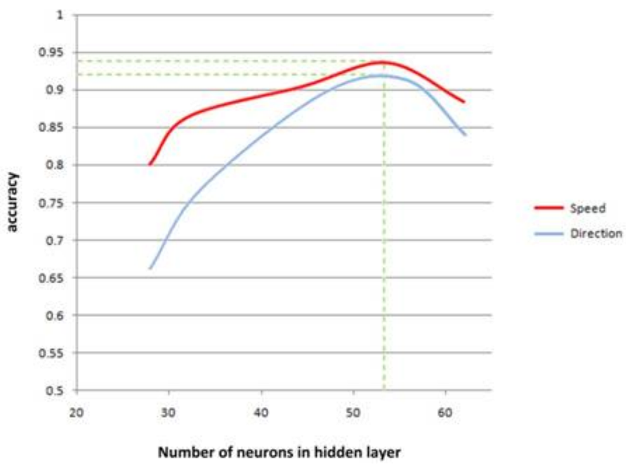


Fig.5. The output accuracy of the ANN in terms of number of neurons in hidden layer

In Figure 5, the output accuracy of the ANN in terms of number of neurons in the hidden layer is shown for the real-time velocity and orientation of the mobile robot. The red curve shows the changes of the velocity and the blue one shows the changes of the orientation of the mobile robot. As the figure, shows, increasing the number of hidden layer neurons from 53, will reduce the output accuracy.

3.2- Obstacle avoidance

In this research, the strategy of obstacle avoidance is a kind of Virtual Potential Field (VPF). When the mobile robot gets near an obstacle, the controller is informed by the rangefinder sensor. Then, images from the camera are analyzed to detect obstacles on the field of view. Detecting edges and colors is the method of image processing. If an obstacle was detected, the approximate size and distance is calculated. Finally, Angle of deviation is determined by using Equation 4.

$$A_{obstacle} = \left(\frac{\sum_{x=\frac{W}{2}-H_{area}}^{\frac{W}{2}+H_{area}} x}{C_{collision} \times W} - 0.5 \right) \times FOV \qquad (4)$$

$A_{obstacle}$ is the angle of deviation in radian. W is width of image received from the rangefinder sensor. H_{area} is the half of the middle of camera image. $C_{collision}$ is the frequency of the obstacle emergence. FOV is the field of view of the rangefinder sensor.

3.3- Virtual parallel paths

After training, in a real operation, the mobile robot has to find the track and moves toward the target position without the need of any human operator. In practice, the mobile robot might go out of the planned trajectory; it may move in an incorrect orientation and lose the trajectory, or maybe face with an obstacle and have a decision to change the path. For this problem, the proposed solution is to train the ANN with some virtual parallel paths that help the mobile robot to return the trajectory.

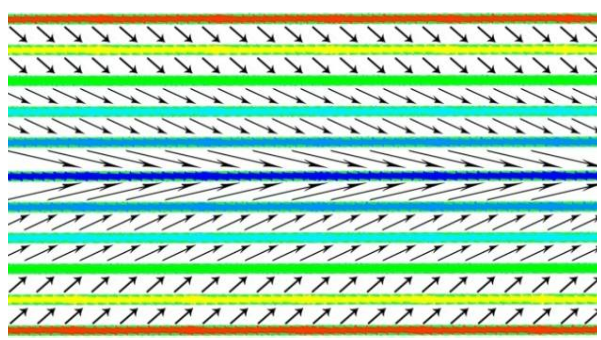


Fig.6. The track is drawn in blue color and virtual parallel paths are in the vicinity

4- Results and Discussions

In the simulation, the mobile robot was moved in the different paths by the human operatormentioned in section 3. While the mobile robot moves toward the target positions, the ANN is being trained. Afterwards, the mobile robot could undertake the task of reaching the target position in aid of the ANN, autonomously. The ANN outputs are the mobile robot orientation and velocity. The outputs of the ANN are discussed in this section.

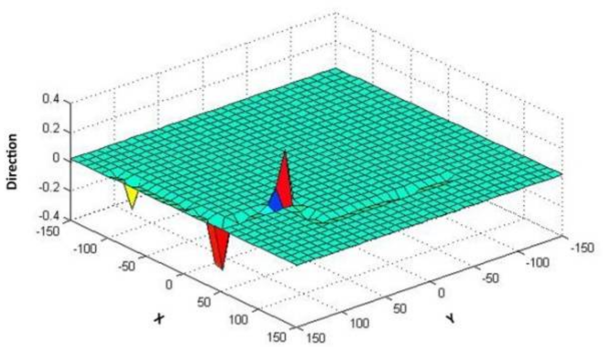


Fig.7. 3D graph of the orientation that extracted from the ANN when the mobile robot autonomously moved on the path: P_1

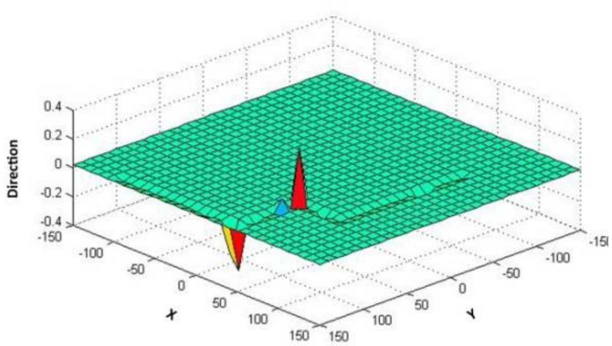
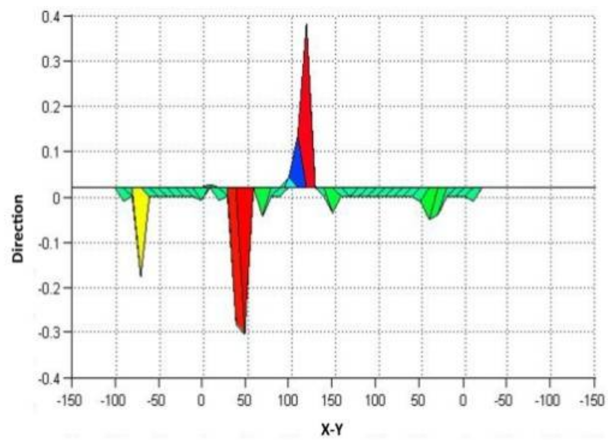
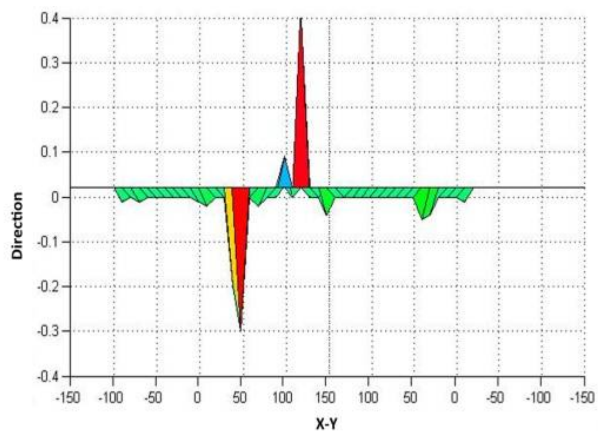


Fig.8. 3D graph of the orientation when the mobile robot is moved in the environment by the human operator

Figure 7 shows the output of the ANN for the orientation of the mobile robot in the environment. The bumps show that the mobile robot has turned right and the sinks show that the mobile robot has turned left. The more the bump height and the sink depth, the greater the orientation value. In Figure 8, the 3D graph of the orientation determined in training process by the human operator is shown.



(a)



(b)

Fig.9. 2D graph of the mobile robot orientation on the path: P_1. (a) The ANN output. (b) Determined by the human operator

For a closer look, Figure 9 shows 2D projections of the graphs displayed in Figures 7, 8. As it can be seen, an error about -0.17 radians that occurs in the range of (-75, 145) caused the mobile robot deviation to the left, but it was covered by the virtual parallel paths strategy and the mobile robot returned the track. This strategy guaranteed that the performance of the extracted orientation from the trained ANN is very similar to the human operator performance.

As depicted in Figure 9 (a), in the range of (-75, 145), an error has occurred in the orientation of -1.8 radians. In the range of (50, 145), the mobile robot has a left turn and its orientation is about -0.3 radians in either navigation by the human operator and the ANN. In the range of (50, 65), the mobile robot has a right turn and its orientation is about 0.4 radians in either navigation by the human operator and the ANN. Also in the other coordinates, the mobile robot navigation by either the human operator or the ANN is almost identical.

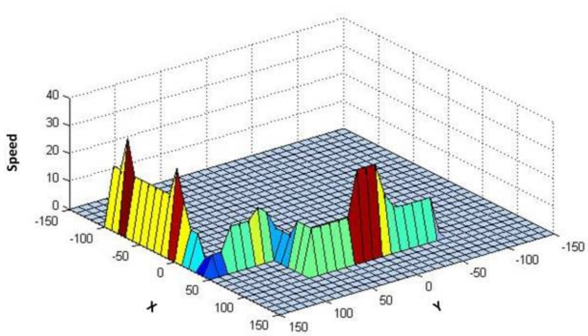


Fig.10. 3D graph of the velocity that extracted from the ANN when the mobile robot autonomously moved on the path: P_1

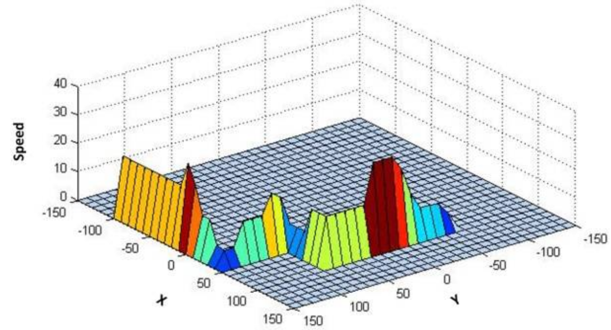


Fig.11. 3D graph of the velocity when the human operator moves the mobile robot in the environment

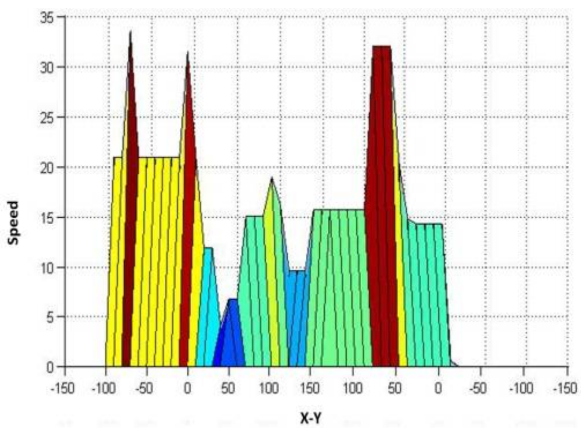
Figure 10 shows the 3D graph of the mobile robot velocity derived by the ANN. The Z-axes shows the absolute value of the mobile robot velocity. The more the bump height, the greater the velocity value. Figure 11 shows the 3D graph of the mobile robot velocity derived by the human operator. The two graphs show very similar performance of the manual system and the ANN.

Table. 2. Standard deviation, Mean, and Maximum of the navigation errors base on experimental results

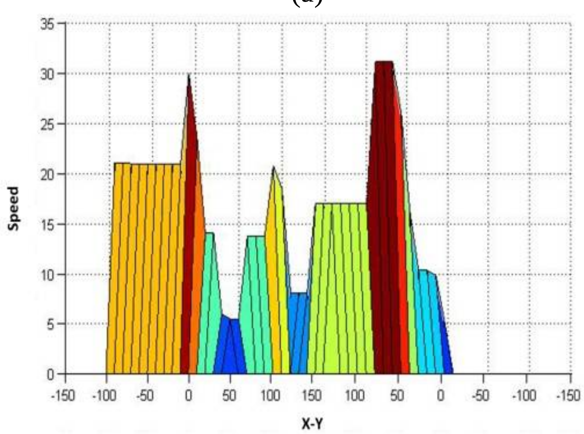
Paths	Orientation error (rad)			Velocity error (Km/H)		
	Std. Dev.	Mean	Max.	Std. Dev.	Mean	Max.
1	0.0361	0.0148	0.1666	2.4068	1.9293	12.5984
2	0.0300	0.0011	0.0869	1.7906	0.8926	7.9433
3	0.0332	0.0141	0.0993	1.9525	1.3479	9.2618
4	0.0424	0.0196	0.1754	2.4095	1.9739	13.0768
5	0.0374	0.0151	0.1697	2.3183	1.6051	12.0672
Tot	0.0346	0.0129	0.1754	2.2592	1.5498	13.0768

Experimental results show the mobile robot has been successful to achieve each target position autonomously. Table 2 , represents error analysis of the mobile robot navigation on five paths.

The total standard deviation of the orientation and velocity errors are 0.0346 radians and 2.2592 Km/h. The mean of errors are 0.0129 radians for the orientation, and 1.5498 Km/h for the velocity.



(a)



(b)

Fig.12. 2D graph of the mobile robot velocity on path: P_1. (a) The ANN output. (b) Determined by the human operator

For a closer look, In Figure 12, by comparing the velocity of the mobile robot in manual and autonomous mode, it can be found that the mobile robot velocity extracted from the ANN is very close to the mobile robot velocity specified by the human operator.

5- Conclusion

In this paper, an ANN based approach is proposed for real-time collision-free navigation of mobile robots. A modified virtual potential field method is used to obstacle avoidance. An unknown environment, which is based on the concept of a city road map, is used for a car-like mobile robot. In a different approach with the previous mobile robot navigating approaches, the ANN inputs are the coordinates of the current position of the mobile robot and the target position, and the ANN outputs are the mobile robot orientation and velocity. Some of the main practical results are:

- The ANN performance in estimating the mobile robot motion parameters is highly depends on the number of neuron in hidden layer. The result shows the best network performance is obtained by increasing the number of hidden layer neurons to maximum of 53 neurons. More neurons will reduce the performance of the ANN.
- The average deviation in estimating correct velocity is.5% and 7.7% for orientation.
- Increasing the number of hidden layer neurons from 53 neuron, will increase the deviation around 2.0%.
- The success of the mobile robot navigation is increased by using the virtual parallel paths. The shorter the specified distance between virtual parallel paths is, the smoother the trajectory of the mobile robot will be.

Results showed the proposed approach is suitable for real-time navigation of a mobile robot in an unknown and partially dynamic environment.

References

[1] Abiyev, R., Ibrahim, D. and Erin, B.,(2010). “Navigation of mobile robots in the presence of obstacles,” Advances in Engineering Software, 41(10): pp. 1179-1186.

[2] Belkhouche, F., Belkhouche, B. and Rastgoufard, P., (2006). “Line of sight robot navigation toward a moving goal,” IEEE Transactions on Systems Man, and Cybernetics B: Cybernetics, 36(2): pp. 255–267.

[3] Chabini, I. and Lan, S., (2002). “Adaptation of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks,” IEEE Transactions on Intelligent Transportation Systems, 3(1): pp. 60–74.

[4] Gu, D. and Hu, H., (2002). “Neural predictive control for a car-like mobile robot,” Robotics and Autonomous Systems, 39: 73–86.

[5] Hui, N. B., Mahendar, V. and Pratihar, D. K., (2006).“Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches,” Fuzzy Sets and Systems, 157: pp. 2171-2204.

[6] Hui, N. B. and Pratihar, D. K., (2009).A comparative study on some navigation schemes of a real robot tackling moving obstacles,” Robotics and Computer-Integrated Manufacturing, 25(4): pp. 810-828.

[7] Kozakiewicz, C. and Ejiri, M., (1991).“Neural Network Approach to Path Planning for Two Dimensional Robot Motion,” International Workshop on Intelligent Robots and Systems, 2: pp. 818-823.

[8] Kumar, S. M., Kumar, M. D., Parhi, D. R.and Prasad, S.M.,(2012).“Intelligent controller for mobile robot based on heuristic rule base network,” Advanced Materials Research, 403: pp. 4777-4785.

[9] Tani, J. and Fukumura, N., (1997). “Self-organising internal representation in learning of navigation:a physical experiment by the mobile robot YAMABICO,” Neural Networks, 10(1): pp. 153–159.

[10] Ueda, M., Iwata, K., Shimizu, T. and Sakai, I., (1975).“Sensors and systems of an industrial robot,” Memoirs of the School of Engineering, Nagoya University, 27(2): pp. 163-207.

[11] Willms, A. R. and Yang, S. X., (2008).“Real-time robot path planning via a distance propagating dynamic system with obstacle clearance,” IEEE Transactions on Systems Man, and Cybernetics B: Cybernetics, 38(3): pp. 884–893.

[12] Yang, S. X. and Meng, M., (2003).“Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach,” IEEE Transactions on Neural Networks, 14: pp. 1541–1552.

[13] Yibin, L. I., Caihong, L. I. and Zijian, Z. H. A. N. G., (2005).“Search strategy of path for mobile robot,” IEEE International Conference on Vehicular Electronics and Safety, pp. 222-227.

[14] Zhu, Q., Hu, J., Cai, W. and Henschen, L., (2011). “A new robot navigation algorithm for dynamic unknown environments based on dynamic path re-computation and an improved scout ant algorithm,” Applied Soft Computing, 11: pp. 4667–4676.